ds_18_coursework

2023-11-30

We will work on a dataset containing a large subset of the human proteome, for each protein you are given its sequence, length, mass, and subcellular location (when available).

First, we import the file:

```r
library(readr)
df <- readr::read_csv("~/Downloads/uniref_merged.csv")
```

```
## New names:
## Rows: 12283 Columns: 24
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (13): Cluster ID, Cluster Name, Types, Organisms, Cluster members, First... dbl
## (5): ...1, Size, Identity, Mass, Length lgl (6): Golgi, Membrane,
## Extracellular, Endoplasmic Reticulum, Nucleus, Cy...
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

```r
df
```

```
## # A tibble: 12,283 x 24
##     ...1 `Cluster ID`    `Cluster Name`           Types  Size Organisms Identity
##    <dbl> <chr>           <chr>                    <chr> <dbl> <chr>        <dbl>
##  1     0 UniRef50_A8MXK1 Cluster: V-set and tran~ UniP~    98 Homo sap~      0.5
##  2     1 UniRef50_F8WCM5 Cluster: Insulin, isofo~ UniP~    19 Homo sap~      0.5
##  3     2 UniRef50_O14713 Cluster: Integrin beta-~ UniP~   373 Homo sap~      0.5
##  4     3 UniRef50_O15347 Cluster: High mobility ~ UniP~   131 Homo sap~      0.5
##  5     4 UniRef50_P0DMR3 Cluster: Putative prote~ UniP~     1 Homo sap~      0.5
##  6     5 UniRef50_P35243 Cluster: Recoverin       UniP~    60 Homo sap~      0.5
##  7     6 UniRef50_P40617 Cluster: ADP-ribosylati~ UniP~   625 Homo sap~      0.5
##  8     7 UniRef50_P61026 Cluster: Ras-related pr~ UniP~   115 Homo sap~      0.5
##  9     8 UniRef50_P61086 Cluster: Ubiquitin-conj~ UniP~   224 Homo sap~      0.5
## 10     9 UniRef50_Q15528 Cluster: Mediator of RN~ UniP~    61 Homo sap~      0.5
## # i 12,273 more rows
## # i 17 more variables: `Cluster members` <chr>, `First member` <chr>,
## #   Entry <chr>, `Entry Name` <chr>, `Protein names` <chr>, `Gene Names` <chr>,
## #   Organism <chr>, Mass <dbl>, Sequence <chr>,
## #   `Gene Ontology (cellular component)` <chr>, Length <dbl>, Golgi <lgl>,
## #   Membrane <lgl>, Extracellular <lgl>, `Endoplasmic Reticulum` <lgl>,
## #   Nucleus <lgl>, Cytoplasm <lgl>
```

Next, we compute the number of each aminoacids from the sequence:

```r
library(stringr)
aminoacids <- unique(strsplit(paste(df$Sequence, sep="", collapse=""), '')[[1]])
# What aminoacids did we find? (You may note something quite interesting here...)
print(aminoacids)
```

```
##  [1] "M" "R" "P" "L" "S" "G" "K" "T" "I" "F" "A" "C" "Q" "V" "Y" "N" "E" "D" "H"
## [20] "W" "U"
```

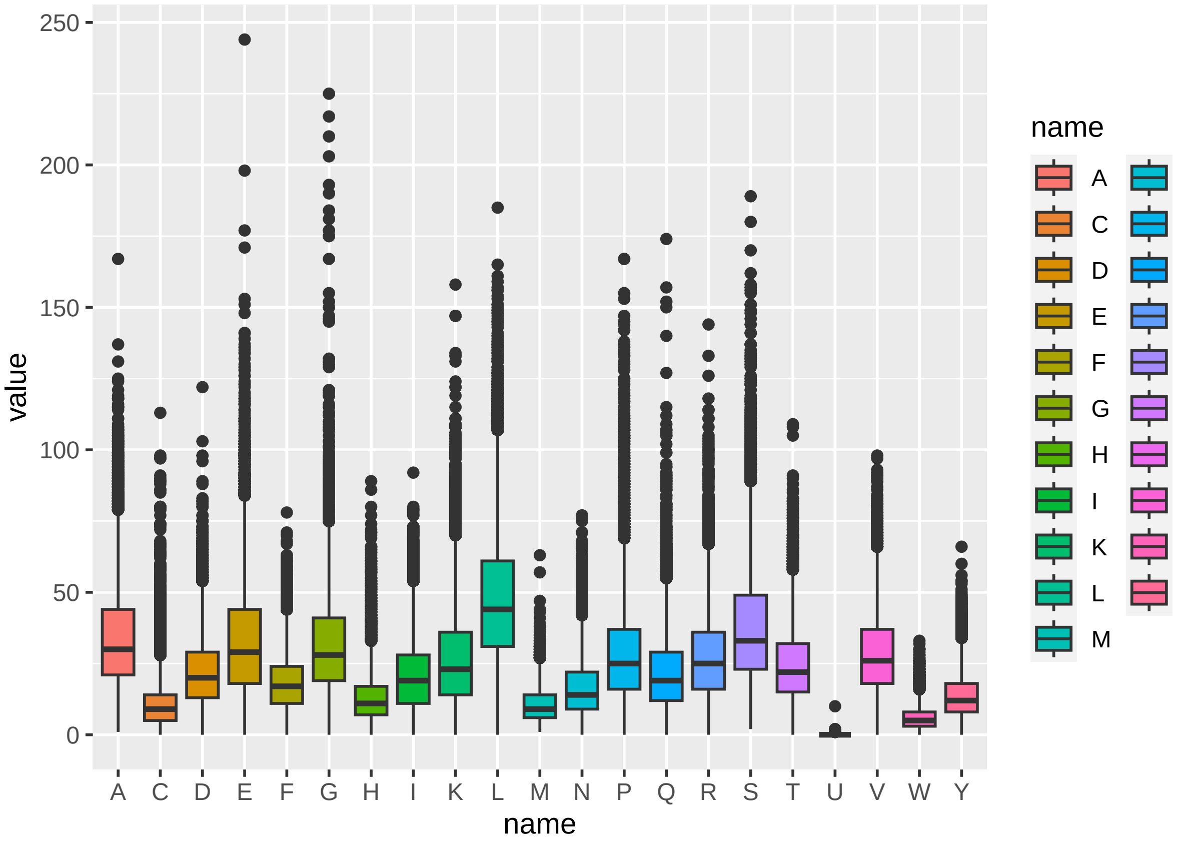
```
for (c in aminoacids) {
  df[c] <- str_count(df$Sequence, c)
}
# Did we miss any amino acids?
all(apply(df[aminoacids], 1, sum) == df$Length)
```

```
## [1] TRUE
```

Let's plot that!

```
library(tidyr)
library(ggplot2)
pivot_longer(df[aminoacids], cols=aminoacids) %>%
ggplot(aes(x=name, y=value, fill=name)) +
  geom_boxplot()
```

```
## Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
## i Please use `all_of()` or `any_of()` instead.
##   # Was:
##   data %>% select(aminoacids)
##
##   # Now:
##   data %>% select(all_of(aminoacids))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



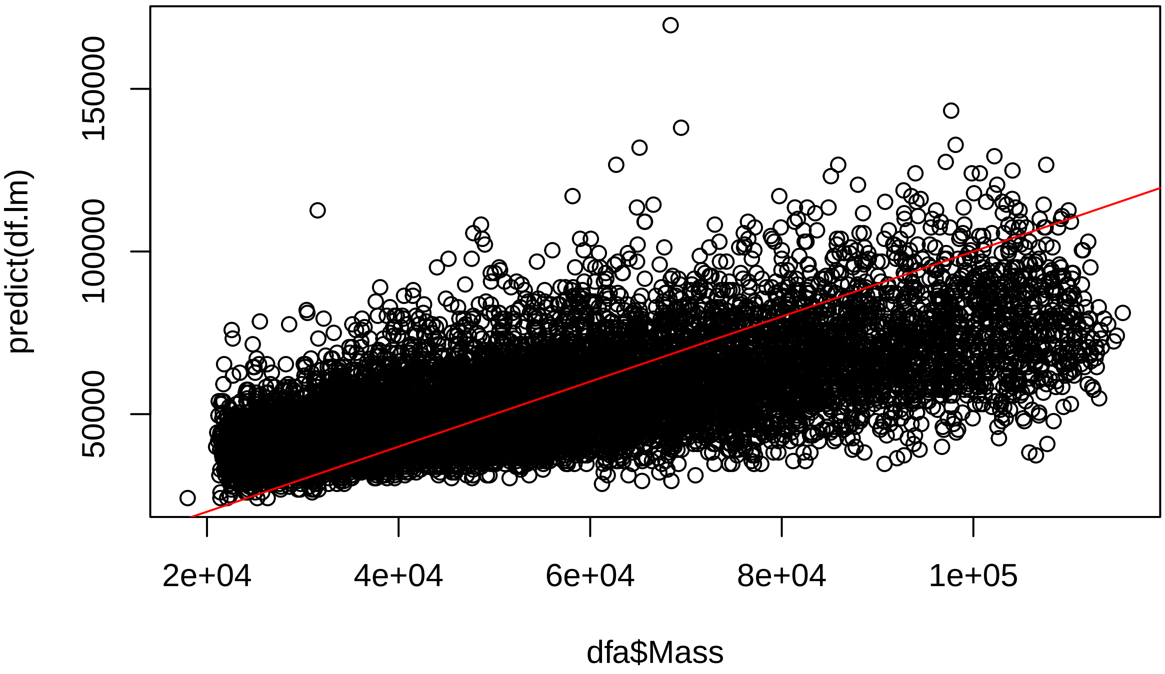
Let's try a simple prediction, the mass as a function of the number of alanines:

```
dfa <- df[c(aminoacids,"Mass")]
df.lm <- lm(Mass ~ A, dfa)
summary(df.lm)
```

```
##
## Call:
## lm(formula = Mass ~ A, data = dfa)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -101179  -10517   -2372    9057   69201
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 23322.978    311.449   74.89   <2e-16 ***
## A             875.772      8.095  108.19   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15970 on 12281 degrees of freedom
## Multiple R-squared:  0.488,  Adjusted R-squared:  0.488
## F-statistic: 1.17e+04 on 1 and 12281 DF,  p-value: < 2.2e-16
```

We can now print the model, and check the coefficients:

```
plot(dfa$Mass, predict(df.lm))
abline(a=1,b=1, col="red")
```



Can you make a near-perfect predictor for the mass, given the aminoacids count? (OK: R-squared > 0.9, Full credit: R-squared > 0.99):

Split your data in a training and test set, and compute a summary statistic for the predictor quality:

Our dataset also contains the subcellular location of proteins, stored as true/false values in the following columns: Golgi, Membrane, Extracellular, Endoplasmic Reticulum, Nucleus, Cytoplasm.

Can you create a plot showing the relationship between sequence length and subcellular location?

Can you create a plot relating bias in sequence composition to subcellular location?

Next, try to make a predictor for the subcellular location, given the sequence. (hint: start with the aminoacid counts!). For full credit, your method should be able to predict multiple possible locations (with a precision greater than 0.2).

Show the results of your predictions using a plot, and appropriate statistics. Are all the categories predicted with the same accuracy? For full credit, the evaluation should be done on a test set separate from your training set.

Finally, export your results. For each predictor, export: - a file with the real values, predicted values, and identifying information for each sample (each column should be appropriately labelled) - a file with summary statistics, across predicted categories. - a plot showcasing the prediction accuracy.

For full credit, create an R shiny app, loading the exported files from the last step, and creating plots from it. Do not attempt this unless you have successfully completed all the previous step!